Amendments to the Specification:

Please replace paragraph [0013] with the following amended paragraph:

[0013] While the hardware is described as having a cache memory system, this cache memory system need not necessarily be the fast cache memory normally used for microprocessors, but instead may ~~simply~~ be random access memory of some kind, or if the FIFO buffer is small enough, a series of registers. Thus, the preferred embodiments of the present invention overcome the problems of the potentially unbounded latency software doorbell signal by allowing the hardware devices to participate, on a limited basis, in the cache coherency protocol using the invalidation signal of the cache coherency system as the notification.

Please replace paragraph [0020] with the following amended paragraph:

[0020] The main memory array 110 preferably couples to the microprocessors 102 and the rest of the computer system 100 through the host node 104. The host node 104 preferably has a memory control unit (not shown) that controls transactions to the main memory array 110 by asserting the necessary control signals during memory accesses. The main memory array 110 ~~generally~~ comprises a ~~conventional~~ memory device or array of memory devices in which program instructions and data may be stored. The main memory array 110 may comprise any suitable type of memory such as dynamic random access memory (DRAM) or any of the various types of DRAM devices such as synchronous (SDRAM), extended data output DRAM (EDO DRAM) or RAMBUS™ DRAM (RDRAM).

Please replace paragraph [0025] with the following amended paragraph:

[0025] In a system having multiple microprocessors 102, and preferably with each microprocessor having at least some cache memory (either internal, L1, cache memory and possibly external, L2, cache memory), there is a need to

~~insure~~ ensure cache coherency across all the caches for all the microprocessors with respect to the main memory, and vice versa. Consider for purposes of example a software stream executed on CPU 102A of Figure 1. Further consider that the software stream updates a variable from main memory, a copy of which is present in the cache of the CPU 102A. By writing a new value to the cache memory location, the cache memory version becomes the only valid version within the system. Cache coherency protocols are responsible for propagating the new value to all the appropriate locations, or at least notifying other CPUs 102 that their copy (if they have one) of the parameter is invalid. While there may be several possible cache coherency protocols that could ensure this coherency, in the preferred embodiments the cache coherency protocol is a write-back invalidate protocol. In a write-back invalidate cache coherency protocol, each agent wanting to modify memory must seek and obtain modify rights prior to the modification of the memory location. In being granted modify rights by the coherency system, other shared copies of the memory location are invalidated. Consider for purposes of explanation a piece of memory shared between two microprocessors. In this initial state, the status of the memory location is shared valid in each microprocessor. Further consider that a first processor seeks to modify the memory location, and thus requests permission from the device implementing cache coherency for this permission, in the preferred embodiments host node 104. The host node 104 grants permission to modify the memory location (gives the requesting microprocessor exclusive ownership), and simultaneously invalidates other copies, in the exemplary case the duplicate held in the second microprocessor. The microprocessor having exclusive ownership may change the value of the memory location (or rather the version stored in the microprocessor's cache) at will. It is not until the memory location is evicted from the requesting processor's cache, or some other device (the second microprocessor or other device capable of running memory transactions) requests the data at the memory location, that the updated value is written back to main memory. A write-back invalidate cache coherency protocol is preferred because write through cache protocols, requiring each cache line modification to

be written back to main memory, are not compatible with the 64 bit machines offered by Intel®.

Please replace paragraph [0028] with the following amended paragraph:

[0028] More particularly, the host node logic 104 preferably has a series of registers 124, 126 and 128. The registers 124, 126, 128 preferably identify the top of the FIFO buffer, the bottom of the FIFO buffer, and a destination respectively. The top 124 and the bottom 126 registers ~~simply~~ indicate the range in main memory of the location of the FIFO buffer 120. Inasmuch as the FIFO buffer 120 is preferably a linearly addressed set of memory locations within the main memory 110, the location may be completely and uniquely identified by having the top address and bottom address. Alternatively, the register 124 could contain a starting address, and register 126 could contain an offset indicating the length of the FIFO buffer. Operation of the registers 124, 126 and 128 of the preferred embodiment is best described with regard to an exemplary write of information to the FIFO buffer 120. In particular, consider a software stream executed on one of the microprocessors, for example microprocessor 102A. Further assume that the microprocessor 102A does not have a copy of the FIFO buffer 120 in its onboard cache (not shown). Thus, the microprocessor 102A copies portions of, or the entire, FIFO buffer from the main memory 110 to be placed in its cache. Further, because the software intends to update these locations, the microprocessor 102A requests of the cache coherency protocol exclusive ownership of those memory locations. Preferably, the host node 104 compares the addresses for which the microprocessor 102A requests exclusive ownership to the top and bottom registers 124 and 126 respectively. Because all or a part of the FIFO buffer 120 is preferably duplicated in the cache memory system in the hardware device 118, this comparison of the addresses to the registers 124 and 126 preferably reveals that the hardware device 118 likewise has copies of these location. The cache coherency protocol preferably simultaneously grants exclusive ownership of the memory locations to the

requesting microprocessor 102A, and invalidates the copies held in the hardware device 118 by sending in an invalidate command to the location identified by the destination register 128.